

# 基于基因水平多样性的微种群教与学优化算法<sup>\*</sup>

王筱薇<sup>1</sup>, 范勤勤<sup>1, 2†</sup>, 王维莉<sup>1</sup>

(1. 上海海事大学 物流研究中心, 上海 201306; 2. 上海交通大学大学 系统控制与信息处理教育部重点实验室, 上海 200240)

**摘要:** 种群多样性对微种群教与学优化算法的性能有极大影响。为进一步提高其性能, 提出一种基于基因水平多样性的微种群教与学优化算法(MTLBO-GLD)。该算法从基因水平上对种群多样性进行监测; 并使用混沌搜索和余弦函数分阶段进行扰动以增加种群多样性。所提算法与 8 种元启发式算法(4 种微种群算法和 4 种非微种群算法)在 13 个测试函数上进行性能比较。实验结果表明, MTLBO-GLD 的整体性能要显著好于其他 8 种所比较的算法。

**关键词:** 教与学优化; 微种群; 混沌搜索; 种群多样性

**中图分类号:** TP      doi: 10.19734/j.issn.1001-3695.2020.03.0050

## Micro-population teaching-learning-based optimization based on gene level diversity

Wang Xiaowei<sup>1</sup>, Fan Qinquin<sup>1, 2†</sup>, Wang Weili<sup>1</sup>

(1. Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China; 2. Key Laboratory of System Control & Information Processing, Ministry of Education of China, Shanghai Jiao Tong University, Shanghai 200240, China)

**Abstract:** The performance of micro-population Teaching-learning-based optimization (TLBO) algorithm is mainly depended on the population diversity. In order to further improve its performance, this paper proposed a Micro-population teaching-learning-based optimization based on gene level diversity (MTLBO-GLD). In the proposed algorithm, the population diversity is monitored at the genetic level, and the chaos search and the cosine function are used to improve the population diversity in different evolutionary stages. The proposed algorithm is compared with eight meta-heuristic algorithms (including four micro-population algorithms and four non-micro-population algorithms) on 13 test functions. The experimental results show that the overall performance of the MTLBO-GLD is significantly better than that of compared algorithms.

**Key words:** teaching-learning-based optimization; micro population; Chaotic search; population diversity

## 0 引言

传统数学方法在求解各类复杂优化问题时存在诸多不足; 因此, 各类智能优化算法被不断提出。如, 遗传算法(genetic algorithm, GA)<sup>[1]</sup>、蚁群算法(ant colony optimization, ACO)<sup>[2]</sup>、粒子群算法(particle swarm optimization, PSO)<sup>[3]</sup>、差分进化算法(differential evolution, DE)<sup>[4]</sup>等。

教与学优化算法(teaching-learning-based optimization, 简称 TLBO)是 Rao 等人<sup>[5,6]</sup>于 2011 年提出的一种新型智能优化算法。其原理主要是模拟学生学习的过程, 该算法分为两个阶段: a)学生在课堂上听老师上课的学习过程; b)学生课下的自我学习过程。由于 TLBO 算法实现简单、便于理解, 且其控制参数较少, 故受到众多学者的重视和研究; 并用其解决了许多实际问题。例如机械优化设计<sup>[6]</sup>、热交换器优化<sup>[7]</sup>、热电制冷器优化<sup>[8]</sup>、平面钢框架的设计优化<sup>[9]</sup>等。

一般来说, TLBO 算法及其他智能优化算法都会使用较大的种群规模来保持种群多样性以提高搜索性能; 但在计算资源受限情况下(如内存小), 较大的种群规模会影响算法运行。为解决以上问题, 基于微种群的智能优化算法被不断提出。例如, Ren 等<sup>[10]</sup>提出一种小种群的差分进化算法(Differential evolution using smaller population, DESP)。该算法不仅提出一种新的 DE 变异策略用于提高全局搜索能力, 还对扰动强度进行自适应调整。实验表明, 与 DE 算法相比,

DESP 算法可以较快寻到最优解。Salehinejad 等<sup>[11]</sup>提出一种基于向量化随机突变因子的差分进化算法(micro-differential evolution with vectorized random mutation factor, MDEV)。该算法将突变因子向量随机化, 即给种群中的每个个体随机分配一个差分突变因子  $F$ , 从而增加种群的多样性。实验结果表明, 与 DE 算法相比, MDEV 算法在搜索过程中能够较好地增强种群的多样性, 同时具有较高的勘探能力。Brown 等<sup>[12]</sup>提出一种使用小种群的自适应差分进化算法(adaptive differential evolution with a small population,  $\mu$ JADE); 该算法的改进是基于 Rcr-JADE 算法<sup>[13]</sup>。所提算法使用一种新的变异算子以提高小种群的全局搜索能力, 并可以加速算法收敛速度。此外,  $\mu$ JADE 还使用隔代更新  $\mu_{CR}$  和  $\mu_F$  的方式来实现 DE 变异因子  $F$  和交叉因子  $CR$  的自适应调整, 从而减少在使用小种群时  $F$  和  $CR$  值迅速减小的发生。研究表明, 尽管  $\mu$ JADE 算法使用了小种群, 但从所选测试函数来看, 其表现比一些使用常规种群规模的 DE 算法更好。Salehinejad 等<sup>[14]</sup>提出一种基于反向学习的集成差分进化算法(opposition-based ensemble micro-differential evolution, OEMDE)。该算法不仅使用反向学习生成与当前个体(候选解)相反的个体以提高算法的全局搜索能力; 在每一代, 各个个体还随机选择一个突变策略以提高种群的多样性。结果表明, 与其他 Micro-DE 算法相比, OEMDE 算法具有更好的性能。

虽然众多学者对微种群优化算法进行了研究, 但是他们

收稿日期: 2020-03-20; 修回日期: 2020-05-12      基金项目: 国家重点研发计划资助项目(2016YFC0800200); 国家自然科学基金资助项目(61603244);

中国博士后科学基金资助项目(2018M642017)

**作者简介:** 王筱薇(1996-), 女, 安徽黄山人, 硕士研究生, 主要研究方向为智能算法; 范勤勤(1986-), 男(通信作者), 浙江宁波人, 副教授, 博导, 博士, 主要研究方向为多目标优化、机器学习、进化计算(forever123fan@163.com); 王维莉(1987-), 女, 湖北武汉人, 讲师, 硕导, 博士, 主要研究方向为行人交通与应急疏散动力学。

的研究大都基于个体水平的种群多样性。在实际种群进化过程中, 如果某个或某几个决策变量过早收敛, 那么就会对个体的进化造成极大影响; 进而影响整个种群进化。基于文献[15]和[16]的研究, 本研究提出一种基于基因水平多样性的微种群教与学优化算法 (micro-population teaching-learning-based optimization based on gene level diversity, MTLBO-GLD)。在所提算法中, 首先利用基因水平检测方法对微种群算法的种群多样性进行维度检测, 然后使用混沌搜索和余弦函数对收敛变量进行分阶段扰动以增强其多样性。最后, 将所提算法与其他 8 种算法在 13 个测试函数上进行性能比较。实验结果表明, MTLBO-GLD 不仅具有较快的收敛速度, 还具有较强全局搜索能力。

## 1 相关算法介绍

### 1.1 标准教与学优化算法

标准 TLBO 算法<sup>[5,6]</sup>主要包含两个阶段, 老师上课的教学过程和学生课余的自学过程。具体算法描述如下。

#### 1.1.1 教学阶段

在教学阶段, 将适应度值最好的个体(设为  $X_T$ )作为这个阶段的教师; 教师希望通过教学过程提高所有学生的平均成绩(设为  $X_M$ )并尽量靠近他自己的水平。该阶段个体的更新公式如下:

$$X_{i,new} = X_{i,old} + r \cdot (X_T - T_F \cdot X_M) \quad (1)$$

其中  $X_{i,new}$  是经过“教”阶段后第  $i$  个学生的新位置。 $T_F = \text{round}(1 + \text{rand}(0,1))$ , 为教学因子, 取 1 或 2,  $r$  为  $[0,1]$  范围内的均匀随机数。若  $X_{i,new}$  的适应度值优于  $X_{i,old}$ , 则更新个体; 否则, 不更新。

#### 1.1.2 学习阶段

在学习阶段中, 学习者主要通过课下与同学交流来提高自己的成绩。个体的更新公式如下:

$$\begin{cases} X_{p,new} = X_{p,old} + r \cdot (X_p - X_l) & \text{if } f(X_p) < f(X_l) \\ X_{p,new} = X_{p,old} + r \cdot (X_l - X_p) & \text{if } f(X_l) < f(X_p) \end{cases} \quad (2)$$

其中,  $p \neq l$ ,  $p, l \in (1, 2, \dots, NP)$ ;  $f(X)$  为适应度函数值。若  $X_{p,new}$  的适应度值优于  $X_{p,old}$ , 则更新个体; 否则, 不更新。

#### 1.1.3 标准 TLBO 算法的实现步骤

- 初始化种群, 设定维度  $D$ 、种群规模  $NP$  和最大评价次数。
- 教学阶段: 班级中的每个学员根据式(1)进行更新个体, 并进行个体选择。
- 学习阶段: 通过式(2)对个体进行更新, 并进行个体选择。
- 判定程序是否达到终止条件, 若没有达到, 则转至步骤 b); 否则, 继续执行。
- 输出最优解。

### 1.2 混沌映射

混沌现象是非线性动力系统的固有特性, 在没有任何一种随机性因子的作用下也能够发生类随机的运动行为, 这种运动行为没有周期的特性不具有收敛特征并且对于初始化的取值有很强的依赖性<sup>[17]</sup>。混沌模型还有一个重要的特征是遍历性, 所以用来对个体进行随机次数的扰动使其跳出局部最优<sup>[18]</sup>。借鉴文献[18]中的混沌扰动策略, 本研究将利用 Logistics 搜索策略来对种群进行基因水平上的扰动, 公式如下:

$$x_{new} = x_{old} \cdot \mu \cdot (1 - x_{old}) \quad (3)$$

其中,  $x_{new}$  是经过混沌映射之后产生的新变量;  $\mu$  为控制参数, 取值范围在  $[0,4]$  之间, 且当  $\mu=4$  时, Logistics 映射是属于完全混沌的状态。

### 1.3 余弦扰动

余弦函数具有比较强的振荡性, 且在  $(0, \pi/2)$  区间中值

呈递减趋势。因此, 它不仅适用于种群的局部扰动, 而且能使算法的进化后期注重局部搜索。根据文献[19]的研究, 本文使用该余弦因子来对个体维度进行扰动, 式子如下:

$$x_{j1,G} = 2 \cdot \cos\left(\frac{\pi}{2} \cdot \frac{FES}{MaxFES}\right) \cdot N(0.5, 0.2) \cdot x_{jbest,G} \quad (4)$$

其中,  $MaxFES$  为设定的最大迭代数;  $N$  表示正态分布函数;  $FES$  为当前迭代数。  $x_{jbest,G}$  为第  $g$  代中多样性最好的维度;  $x_{j1,G}$  为第  $g$  代中第  $j1$  个维度扰动后更新的值,  $j1 \in (1, 2, \dots, D)$ 。

## 2 基于基因水平多样性的微种群教与学优化算法

对于微种群算法来说, 保证种群多样性十分关键。由于微种群算法的种群规模较小, 导致种群多样性会随着环境的选择急剧下降, 这对基于种群进化的元启发式算法来讲十分不利。同时, 已有研究对于种群多样性的评判大都基于个体水平, 如果种群中的个体出现某一维或某几维收敛的情况, 就会对整个个体的进化带来显著影响。因此, 基于文献[15]和[16]的研究, 本研究将对种群进行基因水平上的监测, 并分别使用两种不同的方法来提高种群多样性。

### 2.1 基因水平多样性监测

基于文献[15], 本文采用方差来监测每个变量的多样性, 计算第  $g$  代个体变量的第  $j$  个维度均值和标准差公式如下:

$$m_{j,G} = \frac{1}{NP} \sum_{i=1}^{NP} x_{i,j,G} \quad (5)$$

$$std_{j,G} = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} (x_{i,j,G} - m_{j,G})^2} \quad (6)$$

一般情况下, 若第  $j$  个维度的多样性比较差, 则其  $std_{j,G}$  值也就较小。从理论上来说, 当一个维度的  $std_{j,G}$  值等于 0 时, 表示该维度的多样性完全散失。正常情况下, 元启发式算法在算法的搜索过程中种群多样性是越来越差的。即, 每个维度的  $std_{j,G}$  值都是一个逐渐变小的趋势。因此, 所提算法先从基因水平上来判断每个变量的多样性。需要指出的是, 考虑到如果每代多样性检测会造成计算资源的浪费, 而且也没必要。因此, 本研究采取每 5 代的间隔来进行多样化检测(代数的敏感性分析的实验结果见 3.4.2 节)。

### 2.2 多样性的提高

研究表明  $std_{j,G}$  的值降为 0 需要很长的时间<sup>[15]</sup>, 故本研究用  $o_{j,G}$  来识别种群多样性开始变小的时刻, 式子如下:

$$o_{j,G} = \left(1 - \frac{D \cdot FES}{MaxFES}\right) \cdot a_{j,G} \quad (7)$$

其中

$$a_{j,G} = \max(\mu_{j,G}, 1 - \mu_{j,G}) \quad (8)$$

$$\mu_{j,G} = \frac{m_{j,G} - low_{j,G}}{up_{j,G} - m_{j,G}} \quad (9)$$

$$low_{j,G} = \min(m_{j,G}, x_{low,j}) \quad (10)$$

$$up_{j,G} = \max(m_{j,G}, x_{up,j}) \quad (11)$$

其中,  $x_{low,j}$  和  $x_{up,j}$  分别表示第  $j$  维度预设的上下界。

由式(7)可以知道  $o_{j,G}$  的值会随着迭代次数的增加慢慢变小。一般情况下, 算法种群的多样性也会在迭代过程中慢慢变差, 所以本研究用  $o_{j,G}$  来判断种群各维度的多样性。从式(8)~(11)可以看出,  $a_{j,G}$  主要为  $o_{j,G}$  提供了一些该维度的信息, 以便引导  $o_{j,G}$  更好的判断该维度的多样性。

此外, 本文给  $o_{j,G}$  设置了一个下限值。一般而言, 在算法进化后期, 种群的多样性很小, 但考虑到算法在进化后期会强调局部搜索, 故不再对种群采取扰动措施, 直到算法收敛到最优。在本研究中,  $o_{j,G}$  的最小值设为  $10^{-5}$ (敏感性分析的实验结果见 3.4.3 节)。

如果监测到某个变量的多样性丢失, 那么采用以下方法对其进行扰动来提升多样性。其步骤如下:

- a) 若  $10^{-3} < std_{j,G} < o_{j,G}$ , 则利用式(3)对该维度进行混沌扰动;
- b) 若  $o_{j,G} \leq 10^{-3}$  且  $std_{j,G} < o_{j,G}$ , 则利用式(4)对该维度进行余弦扰动。

2.3 MTLBO-GLD 算法的实现步骤

- a) 初始化: 设定微种群规模  $NP$ 、维度  $D$ 、最大评价次数; 初始化种群;
- b) 教学阶段: 根据式(1)对个体进行更新, 并对个体进行选择;
- c) 学习阶段: 根据式(2)对个体进行更新, 并对个体进行选择;
- d) 每隔 5 代根据式(6)对种群中的每个变量进行多样性检测操作, 并计算出每个维度的标准差, 然后根据式(7)计算该维度相应的  $o_{j,G}$  值, 将两值进行比较;
- e) 根据步骤 d) 进行分阶段扰动: 若  $10^{-3} < std_{j,G} < o_{j,G}$ , 则利用式(3)对该维度进行混沌扰动; 若  $o_{j,G} \leq 10^{-3}$  且  $std_{j,G} < o_{j,G}$ , 则利用式(4)对该维度进行余弦扰动;
- f) 完成步骤 d) 和 e) 后, 对个体进行选择;
- g) 判定算法是否达到设定的终止条件, 若结论是肯定, 则继续执行; 否则, 转回步骤 b) ;
- h) 输出结果。

3 仿真测试

为测试所提算法性能, 本文选取文献[20]中的 13 个测试函数, 并将测试函数的维度设为 30。

将所提算法分别与 TLBO<sup>[5,6]</sup>、DESP<sup>[10]</sup>、MDEVM<sup>[11]</sup>、 $\mu$ JADE<sup>[12]</sup>、Rcr-JADE-s4<sup>[13]</sup>、FSADE<sup>[21]</sup>、TLBO-CSWL<sup>[22]</sup>以及 OBL- $\mu$ TLBO<sup>[23]</sup>等算法进行对比。针对各种算法的种群规模会在以下部分进行说明。所有算法均在 13 个测试函数上独立运行 30 次, 所得结果取 30 次的平均值。同时, 为保证所得结论的可靠性, 仿真结果采用 Friedman<sup>[24]</sup>、Bonferroni-Dunn<sup>[25]</sup>、Holm 和 Hochberg<sup>[26]</sup>检验进行统计分析。显著性水

平设为 5%。表中最优结果加粗表示。

3.1 与微种群算法的比较

在这部分中, 所提算法跟 MDEVM、DESP、 $\mu$ JADE 及 OBL- $\mu$ TLBO 4 种微种群算法进行比较。所有算法的种群规模均设定为 8, 最大适应度函数评价次数为 300 万。对于 13 个测试函数的求解精度有以下规定, 函数  $f_1$  的求解精度为  $10^{-2}$ , 其余 12 个测试函数的求解精度为  $10^{-8}$ 。若算法在达到设定的终止条件时, 所求解还未达到规定的精度, 则表示该算法在此函数上求解失败, 结果以“-”来表示。仿真结果如表 1 所示。

由表 1 可知, MTLBO-GLD 算法对于所有测试函数均能在终止条件范围内寻到最优解, 且在各个测试函数上的求解结果都显著好于所比较算法, 尤其在  $f_1$ 、 $f_2$ 、 $f_3$ 、 $f_4$ 、 $f_6$ 、 $f_{10}$ 、 $f_{11}$  这几个测试函数上寻优性能非常好。其主要原因在于, 对于 MDEVM、DESP 及  $\mu$ JADE 这三个算法来说, 虽然 DE 算法的全局搜索能力要好于 TLBO 算法, 但所提算法通过从基因水平上对种群进行监测, 并采用混沌与余弦函数分阶段来对种群进行扰动。对于 OBL- $\mu$ TLBO 算法来说, 虽然该算法使用反向学习来提高其全局搜索能力, 但并未从基因水平上对种群多样性进行监测。因此, 相较于 MDEVM、DESP、 $\mu$ JADE 及 OBL- $\mu$ TLBO 4 种微种群算法, 所提算法在该函数测试集上的表现更好。

此外, 本文还使用非参数统计分析方法来对几种算法所求解的平均值进行深入分析, 结果见图 1 和表 2。从图 1 中可知 MTLBO-GLD 算法的寻优效率是最高的, 比其中表现较好的 OBL- $\mu$ TLBO 算法略胜一筹, 更是显著高于其他几个比较算法。另外, 从表 2 也可以看出所提算法的寻优效率要显著优于所比较的几个算法。

综上所述, MTLBO-GLD 算法在基因水平上对种群多样性进行监控, 能够得到更高的求解精度。

表 1 微种群算法的实验结果

Tab. 1 Experimental results of Micro-population algorithms

函数		MDEVM	DESP	$\mu$ JADE	OBL- $\mu$ TLBO	MTLBO-GLD
$f_1$	平均值	2.8E+04	1.9E+04	2.2E+04	2.6E+02	<b>0.0E+00</b>
	标准方差	3.3E+03	8.0E+02	7.8E+02	8.9E+00	<b>0.0E+00</b>
$f_2$	平均值	3.1E+04	-	3.7E+04	3.9E+02	<b>0.0E+00</b>
	标准方差	4.7E+03	-	1.2E+03	1.1E+01	<b>0.0E+00</b>
$f_3$	平均值	1.4E+05	-	1.6E+05	4.3E+02	<b>0.0E+00</b>
	标准方差	1.5E+04	-	7.7E+03	5.3E+01	<b>0.0E+00</b>
$f_4$	平均值	-	-	2.2E+05	4.3E+02	<b>0.0E+00</b>
	标准方差	-	-	1.6E+04	1.2E+01	<b>0.0E+00</b>
$f_5$	平均值	2.3E+06	-	2.1E+05	1.8E+05	<b>2.5E+01</b>
	标准方差	4.0E+05	-	5.5E+04	4.3E+04	<b>1.2E+00</b>
$f_6$	平均值	1.8E+04	-	1.2E+04	1.2E+02	<b>0.0E+00</b>
	标准方差	6.3E+03	-	3.8E+03	1.0E+01	<b>0.0E+00</b>
$f_7$	平均值	8.9E+05	-	2.3E+05	2.1E+03	<b>2.5E-06</b>
	标准方差	7.1E+05	-	1.6E+05	8.1E+02	<b>1.8E-06</b>
$f_8$	平均值	-	-	1.0E+05	1.9E+05	<b>3.8E+03</b>
	标准方差	-	-	5.1E+03	6.7E+03	<b>1.4E+03</b>
$f_9$	平均值	-	-	1.2E+05	2.7E+02	<b>0.0E+00</b>
	标准方差	-	-	6.8E+03	1.3E+01	<b>0.0E+00</b>
$f_{10}$	平均值	-	-	3.8E+04	3.9E+02	<b>0.0E+00</b>
	标准方差	-	-	5.0E+03	1.0E+01	<b>0.0E+00</b>
$f_{11}$	平均值	2.6E+04	3.1E+04	4.6E+04	2.8E+02	<b>0.0E+00</b>
	标准方差	3.1E+03	1.7E+03	3.7E+04	1.3E+01	<b>0.0E+00</b>
$f_{12}$	平均值	6.6E+04	1.8E+06	3.8E+04	4.1E+03	<b>6.9E-05</b>
	标准方差	6.5E+04	0.0E+00	7.5E+03	7.8E+02	<b>1.9E-04</b>
$f_{13}$	平均值	7.0E+04	-	3.2E+04	2.5E+05	<b>1.1E-02</b>
	标准方差	6.4E+04	-	1.8E+04	4.8E+05	<b>1.2E-02</b>



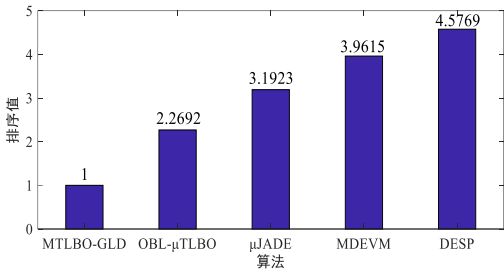


图 1 Friedman 测试在微种群算法所得排序结果

Fig. 1 Ranking obtained by Friedman's test for micro-population algorithms

表 2 Bonferroni-Dunn、Holm 和 Hochberg 在微种群算法上所得  $p$  值

Tab. 2  $P$ -values obtained by Bonferroni-Dunn's, Holm's, and

表 3 所提算法与非微种群算法的结果

Tab. 3 Experimental results of the proposed algorithm and non-Micro-population algorithms

函数		FSADE ( $NP=50$ )	$R_{cr}$ -JADE-s4 ( $NP=100$ )	TLBO ( $NP=30$ )	TLBO-CSWL ( $NP=40$ )	MTLBO-GLD ( $NP=8$ )
$f_1$	平均值	3.5E+04	2.1E+04	4.3E+03	1.9E+03	<b>0.0E+00</b>
	标准方差	5.3E+02	4.9E+02	9.1E+02	4.2E+01	<b>0.0E+00</b>
$f_2$	平均值	4.5E+04	3.3E+04	7.6E+03	2.8E+03	<b>0.0E+00</b>
	标准方差	5.7E+02	7.6E+02	1.3E+02	4.6E+01	<b>0.0E+00</b>
$f_3$	平均值	2.6E+06	6.7E+04	1.7E+04	2.9E+03	<b>0.0E+00</b>
	标准方差	4.8E+05	3.6E+03	8.6E+02	1.6E+02	<b>0.0E+00</b>
$f_4$	平均值	2.2E+05	1.0E+05	7.2E+03	3.2E+03	<b>0.0E+00</b>
	标准方差	4.2E+03	3.5E+04	1.5E+02	5.1E+01	<b>0.0E+00</b>
$f_5$	平均值	2.8E+06	7.8E+04	-	4.1E+05	<b>2.5E+01</b>
	标准方差	6.1E+04	2.7E+03	-	5.8E+04	<b>1.2E+00</b>
$f_6$	平均值	1.3E+04	8.8E+03	1.7E+03	1.1E+03	<b>0.0E+00</b>
	标准方差	4.7E+02	3.6E+02	2.2E+02	2.1E+02	<b>0.0E+00</b>
$f_7$	平均值	1.1E+05	2.2E+04	-	2.2E+04	<b>2.5E-06</b>
	标准方差	2.6E+04	9.8E+03	-	6.4E+03	<b>1.8E-06</b>
$f_8$	平均值	4.4E+04	7.0E+04	-	4.5E+05	<b>3.8E+03</b>
	标准方差	1.8E+03	2.8E+03	-	8.2E+03	<b>1.4E+03</b>
$f_9$	平均值	8.9E+04	9.1E+04	-	7.9E+03	<b>0.0E+00</b>
	标准方差	4.8E+03	1.3E+03	-	8.6E+03	<b>0.0E+00</b>
$f_{10}$	平均值	4.8E+04	3.1E+04	2.3E+04	2.8E+03	<b>0.0E+00</b>
	标准方差	5.0E+02	6.4E+02	1.3E+02	4.4E+01	<b>0.0E+00</b>
$f_{11}$	平均值	3.9E+04	2.2E+04	4.1E+03	2.1E+03	<b>0.0E+00</b>
	标准方差	3.7E+03	6.4E+02	3.9E+02	5.8E+02	<b>0.0E+00</b>
$f_{12}$	平均值	4.0E+04	1.9E+04	-	5.0E+04	<b>6.9E-05</b>
	标准方差	1.0E+03	5.5E+02	-	2.5E+03	<b>1.9E-04</b>
$f_{13}$	平均值	4.1E+04	2.1E+04	-	3.8E+05	<b>1.1E-02</b>
	标准方差	1.1E+03	4.9E+02	-	6.1E+04	<b>1.2E-02</b>

从表 3 中可以看出, MTLBO-GLD 算法在 13 个测试函数上所得结果均好于其他几个非微种群算法。因此, 所提算法虽然使用的是较小规模的种群, 但它仍然表现出非常好的寻优性能。原因主要有以下两点: 首先, 所提算法从基因水平上对种群的多样性进行监测, 从而可以从种群中提取更多有效信息; 其次, 采取了分阶段对种群进行扰动, 从而可以保持算法种群的多样性。因此, 算法的寻优性能得到了很大的提升。

为使实验结论更加可靠, 还使用非参数的统计方法来对所有算法的均值进行分析。统计分析结果见图 2 和表 4。由图 2 可知, 与使用常规规模种群的算法相比, MTLBO-GLD 的寻优性能排名第一。从表 4 可以看出, MTLBO-GLD 算法的求解性能要显著性好于所比较的所有非微种群算法。

综上所述, 所提算法虽然为微种群算法, 但其寻优性能并不比常规种群规模的算法差, 反而更优。这恰好证明从基

Hochberg's procedures for Micro-population algorithms					
MTLBO-GLD vs.	z	Unadjusted p	Bonferroni-Dunn p	Holm p	Hochberg p
DESP	5.77	8.04E-09	3.22E-08	3.22E-08	3.22E-08
MDEVM	4.78	1.79E-06	7.18E-06	5.38E-06	5.38E-06
$\mu$ JADE	3.53	4.08E-04	1.63E-03	8.16E-04	8.16E-04
OBL- $\mu$ TLBO	2.05	4.07E-02	1.63E-01	4.07E-02	4.07E-02

### 3.2 与非微种群算法的比较

在这部分中, 所提算法跟 FSADE、Rcr-JADE-s4、TLBO 和 TLBO-CSWL 这 4 种非微种群算法的性能进行比较。期望求解精度同 3.1 部分。四个算法的种群规模设定

依次为 50、100、30 和 40, 算法参数设置均与文献[12]一致。结果见表 3。

因水平上对种群进行监控是一种十分有效的方法。

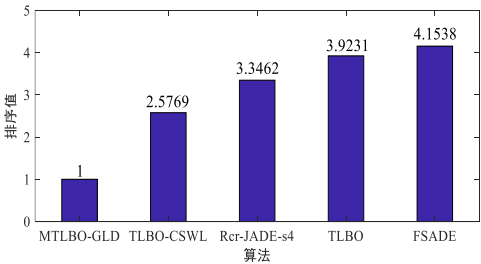


图 2 Friedman 测试在 MTLBO-GLD 和非微种群算法上所得排序结果

Fig. 2 Ranking obtained by Friedman's test for MTLBO-GLD and non-micro-population algorithms

### 3.3 MTLBO-GLD 与 OBL- $\mu$ TLBO、TLBO-CSWL 的比较

为了进一步验证在资源受限情况下的 MTLBO-GLD 算法性能, 这里本文将对所提算法与前面两部分中表现较好的

OBL- $\mu$  TLBO 和 TLBO-CSWL 进行性能比较。与前面部分不同的是,在本实验中,最大的适应度函数评价次数设为 30,000。同时, OBL- $\mu$  TLBO 和 TLBO-CSWL 的种群规模分别设为 8 和 40。所得结果见表 5。从表 5 可以看出 MTLBO-GLD 算法在  $f_1$ 、 $f_2$ 、 $f_3$ 、 $f_4$ 、 $f_5$ 、 $f_6$ 、 $f_{10}$ 、 $f_{11}$ 、 $f_{13}$  上得到的解比 TLBO-CSWL 算法的解要好很多。而 TLBO-CSWL 只在  $f_8$  这个函数上要显著好于所提算法。结果足以表明所提算法的性能是好于 TLBO-CSWL 算法。对于 OBL- $\mu$ TLBO, 两个算法在这些测试函数上得到的平均结果总体上来说差不多, 但所提算法在每个测试函数上表现的比 OBL- $\mu$ TLBO 略胜一筹。因此, MTLBO-GLD 不仅具有较快的收敛速度, 还能够降低计算成本, 同时也展现了较强的全局搜索能力。

本部分还使用非参数统计分析方法来对三种算法的平均值进行分析, 其结果见表 6。由表 6 可知 MTLBO-GLD 算法的整体性能优于 TLBO-CSWL 算法。另外, 虽然 MTLBO-GLD、TLBO-CSWL、OBL- $\mu$ TLBO 的性能在统计学意义上不存在明显的差异, 但是从表 6 中的结果可知, MTLBO-GLD 的整体性能仍稍微优于 OBL- $\mu$ TLBO 和 TLBO-CSWL 的。

表 4 Bonferroni-Dunn、Holm 和 Hochberg  
在 MTLBO-GLD 和非微种群算法上所得  $p$  值

Tab. 4  $P$ -values obtained by Bonferroni-Dunn's, Holm's, and Hochberg's procedures for MTLBO-GLD and non-micro-population algorithms

MTLBO-GLD vs.	$z$	Unadjusted $p$	Bonferroni-Dunn $p$	Holm $p$	Hochberg $p$
FSADE	5.0854	3.67E-07	1.47E-06	1.47E-06	1.47E-06
TLBO	4.7133	2.44E-06	9.75E-06	7.31E-06	7.31E-06
$R_{cr}$ -JADE-s4	3.7831	1.55E-04	6.20E-04	3.10E-04	3.10E-04
TLBO-CSWL	2.0225	4.97E-02	0.2182	4.97E-02	4.97E-02
TLBO-CSWL	2.5427	1.10E-02	4.40E-02	1.10E-02	1.10E-02

表 5 所有比较算法在 30 维函数上所得结果

Tab. 5 Experimental results of all compared algorithms on 30D functions

函数		TLBO-CSWL	OBL- $\mu$ TLBO	MTLBO-GLD
		(NP=40)	(NP=8)	(NP=8)
$f_1$	平均值	4.0E-223	<b>0.0E+00</b>	<b>0.0E+00</b>
	标准方差	0.0E+00	<b>0.0E+00</b>	<b>0.0E+00</b>
$f_2$	平均值	9.8E-141	<b>0.0E+00</b>	<b>0.0E+00</b>
	标准方差	6.5E-140	<b>0.0E+00</b>	<b>0.0E+00</b>
$f_3$	平均值	2.8E-221	<b>0.0E+00</b>	<b>0.0E+00</b>
	标准方差	0.0E+00	<b>0.0E+00</b>	<b>0.0E+00</b>
$f_4$	平均值	5.5E-224	<b>0.0E+00</b>	<b>0.0E+00</b>
	标准方差	0.0E+00	<b>0.0E+00</b>	<b>0.0E+00</b>
$f_5$	平均值	2.8E+01	2.8E+01	<b>2.6E+01</b>
	标准方差	4.1E-01	2.78E-01	<b>3.5E-01</b>
$f_6$	平均值	9.2E-16	<b>0.0E+00</b>	<b>0.0E+00</b>
	标准方差	1.5E-15	<b>0.0E+00</b>	<b>0.0E+00</b>
$f_7$	平均值	<b>0.0E+00</b>	4.4E-05	2.5E-04
	标准方差	<b>0.0E+00</b>	3.2E-05	1.4E-04
$f_8$	平均值	<b>0.0E+00</b>	6.4E+03	4.0E+03
	标准方差	<b>0.0E+00</b>	1.3E+03	1.5E+03
$f_9$	平均值	<b>0.0E+00</b>	<b>0.0E+00</b>	9.7E-01
	标准方差	<b>0.0E+00</b>	<b>0.0E+00</b>	2.5E+00
$f_{10}$	平均值	4.9E+03	2.1E-16	<b>0.0E+00</b>
	标准方差	1.1E+03	8.5E-16	<b>0.0E+00</b>
$f_{11}$	平均值	<b>0.0E+00</b>	<b>0.0E+00</b>	<b>0.0E+00</b>
	标准方差	<b>0.0E+00</b>	<b>0.0E+00</b>	<b>0.0E+00</b>
$f_{12}$	平均值	<b>3.2E-219</b>	8.3E-02	1.9E-04
	标准方差	<b>0.0E+00</b>	9.7E-02	2.8E-04
$f_{13}$	平均值	2.9E+01	2.6E+00	<b>3.1E-03</b>
	标准方差	6.30E-02	5.3E-01	<b>6.2E-03</b>

表 6 所有比较算法的 Bonferroni-Dunn、Holm 以及 Hochberg 检验的  $p$  值  
Tab. 6  $P$ -Values obtained by Bonferroni-Dunn's, Holm's, and Hochberg's procedures on experimental results with all comparison algorithms

MTLBO-GLD vs.	$z$	Unadjusted $p$	Bonferroni-Dunn $p$	Holm $p$	Hochberg $p$
TLBO-CSWL	1.86	6.24E-02	1.25E-01	1.25E-01	1.25E-01
OBL- $\mu$ TLBO	0.49	6.24E-01	1.00E+00	6.24E-01	6.24E-01

3.4 实验分析

3.4.1 混沌和余弦两种扰动的结果分析

一般而言, 算法在前期应偏重于全局搜索, 在后期应偏重于局部搜索<sup>[27]</sup>。在本研究中, 所提算法采用混沌搜索和余弦扰动两种方法进行分阶段增强种群多样性。为验证混沌搜索和余弦扰动两种方法的特点, 本部分给出两种方法在二维空间中的个体位置变化图。这里以 8 个粒子为例, 粒子迭代 60 次的位置变化图如图 3 和 4, 从图 3 和 4 中可以看出, 混沌搜索的全局搜索能力要好于余弦函数。

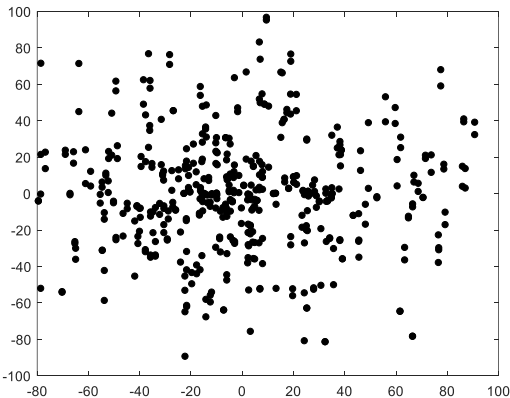


图 3 混沌搜索

Fig. 3 Chaotic search

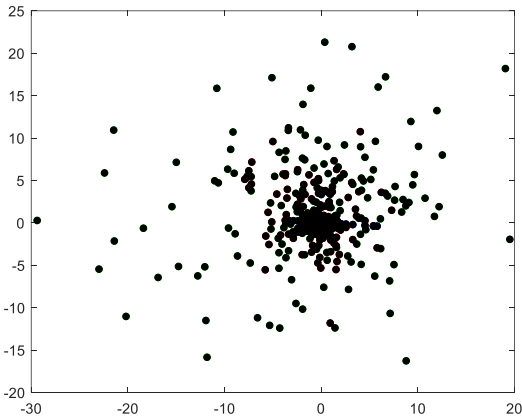


图 4 余弦扰动

Fig. 4 Cosine perturbation

3.4.2 间隔代数敏感性分析

本部分对间隔的代数值进行敏感性分析。本实验中, 种群规模设置为 8, 最大的适应度函数评价次数设为 30,000, MTLBO-GLD 算法独立运行 30 次; 并选取 13 个测试函数。间隔代数分别设定为 0、5、10、15 和 20, 并使用非参数的统计方法对平均值进行排序分析。各个参数的排序结果如图 5 所示。由图 5 可知, 当间隔代数值取 5 时, 所提算法的整体性能表现最佳; 所以间隔的代数值设为 5。

3.4.3  $o_{j,G}$  值敏感性分析

该部分对  $o_{j,G}$  的下限值进行敏感性分析, 实验设置与 3.4.2 相同。 $o_{j,G}$  分别设定为  $10^{-1}$ 、 $10^{-2}$ 、 $10^{-3}$ 、 $10^{-4}$  和  $10^{-5}$ , 不同参数下的排序结果见图 6。由图 6 可知, 在  $o_{j,G}$  的下限设为  $10^{-5}$  时, 所提算法的整体性能最佳。因此, 在所提算法中,  $o_{j,G}$  的下限值设为  $10^{-5}$ 。

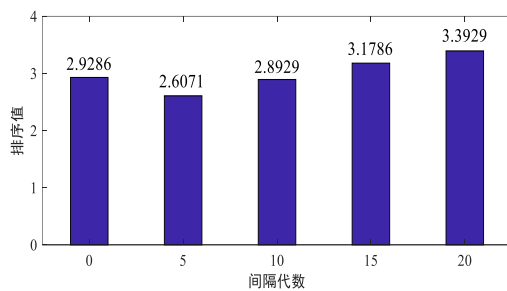


图 5 Friedman 测试在不同代数间隔下所得排序结果

Fig. 5 Ranking obtained by Friedman's test for different iteration intervals

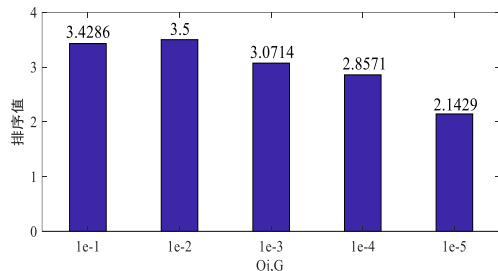


图 6 Friedman 测试在 oj,G 取不同下限值下所得排序结果

Fig. 6 Ranking obtained by Friedman's test for oj,G under different lower limits

## 4 结束语

为平衡微种群教与学优化算法的全局和局部搜索能力, 本文提出一种基于基因水平多样性的微种群教与学 (MTLBO-GLD) 算法。在该算法中, 首先从基因水平上对种群进行监测, 以防止某一维或某几维过早收敛而对整个种群搜索带来不利影响; 然后使用混沌搜索和余弦函数分阶段实现扰动来实现种群多样性。所提算法跟  $\mu$ JADE、DESP、MDEV、OBL- $\mu$ TLBO 等微种群算法和 FSADE、Rcr-JADE-s4、TLBO、TLBO-CSWL 等非微种群算法进行性能比较。实验结果表明 MTLBO-GLD 算法的整体性能要优于所有比较算法。

## 参考文献:

- [1] Goldberg D E. Genetic algorithms in search, optimization, and machine learning, addison-wesley, reading, ma, 1989 [J]. NN Schraudolph and J, 1989, 3 (1) .
- [2] Dorigo M, Maniezzo V, Colomni A. The ant system: optimization by a colony of cooperating agents [J]. IEEE Trans on Systems, Man, and Cybernetics B, 1996, 26 (1): 29-41.
- [3] Eberhart R, Kennedy J. A new optimizer using particle swarm theory [C]// MHS'95. Proc of the 6th International Symposium on Micro Machine and Human Science. IEEE, 1995: 39-43.
- [4] Efren M M, Miranda-Varela M E, Rubi D C G. Differential evolution in constrained numerical optimization: an empirical study [J]. Information sciences, 2010, 180 (22): 4223-4262.
- [5] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems [J]. Information Sciences, 2012, 183 (1): 1-15.
- [6] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems [J]. Computer-aided Design, 2011, 43 (3): 303-315.
- [7] Rao R V, Patel V. Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm [J]. Applied Mathematical Modeling, 2013, 37 (3): 1147-1162.
- [8] Huang Runqing, Xi Lifeng, Li Xinglin, *et al.* Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods [J]. Mechanical Systems & Signal Processing, 2007, 21 (1): 193-207.
- [9] Djuricpm P M, Huang Yufei, Ghirmai T. Perfect sampling: a review and applications to signal processing [J]. IEEE Trans on Signal Processing, 2002, 50 (2): 345-356.
- [10] Ren Xuan, Chen Zhizhao, Ma Zhen. Differential evolution using smaller population [C]// Proc of the 2th International Conference on Machine Learning and Computing. IEEE, 2010: 76-80.
- [11] Salehinejad H, Rahnamayan S, Tizhoosh H R, *et al.* Micro-differential evolution with vectorized random mutation factor [C]// Proc of IEEE Congress on Evolutionary Computation (CEC) . IEEE, 2014: 2055-2062.
- [12] Brown C, Jin Yaochu, Leach M, *et al.*  $\mu$ JADE: adaptive differential evolution with a small population [J]. Soft Computing, 2016, 20 (10): 4111-4120.
- [13] Gong Wenyin, Cai Zhihua, Wang Yang. Repairing the crossover rate in adaptive differential evolution [J]. Applied Soft Computing Journal, 2014, 15 (2): 149-168.
- [14] Salehinejad H, Rahnamayan S, Tizhoosh H R. Opposition-based ensemble micro-differential evolution [C]// Proc of IEEE Symposium Series on Computational Intelligence (SSCI) . IEEE, 2017: 1-8.
- [15] Yang Ming, Li Changhe, Cai Zhihua, *et al.* Differential evolution with auto-enhanced population diversity [J]. IEEE Trans on Cybernetics, 2014, 45 (2): 302.
- [16] Fan Qinqin, Yan Xuefeng. Self-adaptive differential evolution algorithm with discrete mutation control parameters [J]. Expert Systems with Applications, 2015, 42 (3): 1551-1572.
- [17] 孟红记, 郑鹏, 梅国晖, 等. 基于混沌序列的粒子群优化算法 [J]. 控制与决策, 2006, 21 (3): 263-266. (Meng Hongji, Zheng Peng, Mei Guohui, *et al.* A particle swarm optimization algorithm based on chaotic sequence [J]. Control and Decision, 2006, 21 (2): 263-266.)
- [18] 唐娜, 张公让, 李磊, 等. 自适应混沌搜索的双粒子群优化算法 [J]. 计算机工程与设计, 2016, 37 (9): 2421-2428. (Tang Na, Zhang Gongrang, Li Lei, *et al.* Double particle swarm optimization algorithm of self-adaption and chaos search [J]. Computer Engineering and Design, 2016, 37 (9): 2421-2428.)
- [19] 黄宝清, 李俊兴, 宋春宁, 等. 基于余弦控制因子和多项式变异的鲸鱼优化算法 [J]. 控制与决策. (2018-11-13) [2020-03-18]. <http://doi:10.13195/j.kzyjc.2018.0463>. (Huang Baoqing, Li Junxing, Song Chunning, *et al.* Whale optimization algorithm based on cosine control factor and polynomial mutation [J]. Control and Decision. (2018-11-13) [2020-03-18]. <http://doi:10.13195/j.kzyjc.2018.0463>.)
- [20] Yao Xin, Liu Yong. Evolutionary programming made faster [J]. IEEE Trans on Evolutionary Computation, 1999, 3 (2): 82-102.
- [21] Sharma H, Shrivastava P, Bansal J C, *et al.* Fitness based self-adaptive differential evolution [M]// Nature inspired cooperative strategies for optimization (NISCO 2013) . Springer international publishing, 2014: 71-84.
- [22] 柳缔西子, 范勤勤, 胡志华. 基于混沌搜索和权重学习的教与学优化算法及其应用 [J]. 智能系统学报, 2018, 13 (05): 818-828. (Lin Dixizi, Fan Qinqin, Hu Zhihua. Teaching-learning-based optimization algorithm based on chaotic search and weighted learning and its application [J]. CAAI Trans on Intelligent Systems, 2018, 13 (05): 818-828.)
- [23] 范勤勤, 柳缔西子, 王筱薇, 等. 基于反向学习的微种群教与学优化算法及其应用 [J]. 郑州大学学报. (2019-12-27) [2020-03-18]. <http://doi:10.13705/j.issn.1671-6833.2020.01.020>. (Fan Qinqin, Liu Dixizi, Wang Xiaowei, *et al.* Opposition-based learning teaching-learning-based optimization algorithm with a micro population and its

- application [J]. Journal of Zhenzhou University. (2019-12-27) [2020-03-18]. <http://doi: 10.13705/j.issn.1671-6833.2020.01.020>.)
- [24] Milton F. The use of ranks to avoid the assumption of normality implicit in the analysis of variance [J]. Journal of the American Statistical Association, 1937, 34 (200): 109-109.
- [25] Dunn O J. Multiple comparisons among means [J]. Journal of the American Statistical Association, 1961, 56 (293): 52-64.
- [26] Salvador García, Molina D, Lozano M, *et al.* A study on the use of non-parametric tests for analyzing the evolutionary algorithms'behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization [J]. Journal of Heuristics, 2009, 15 (6): 617.
- [27] Fan Qinqin, Yan Xuefeng, Xue Yu. Prior knowledge guided differential evolution [J]. Soft Computing, 2017, 21 (22): 6841-6858.